REMARKS

Claims 1-23, 26 and 27 are currently pending in the subject application and are presently under consideration. Claims 1, 21, 26 and 27 have been amended as shown on pp. 2-6 of the Reply. Applicants' representative thanks Examiner Lee for the courtesies extended during the interview conducted on February 10, 2009 with Evan Perry (Reg. No. 62, 190). Distinctions between the claims and the cited art were discussed; however, no agreement was reached. Favorable reconsideration of the subject patent application is respectfully requested in view of the comments and amendments herein.

I.   **Rejection of Claims 1-6, 8-18 and 27 Under 35 U.S.C. §103(a)**

Claims 1-6, 8-18 and 27 stand rejected under 35 U.S.C. §103(a) in view of Dandoy (U.S. Publication 2004/0230954). It is respectfully submitted that this rejection be withdrawn for at least the following reasons. Dandoy does not teach or suggest each and every feature of the subject claims.

> [T]he prior art reference (or references when combined) must teach or suggest all claim limitations. *See* MPEP §706.02(j). *See In re Vaeck,* 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991). *See In re Royka,* 490 F.2d 981, 180 USPQ 580 (CCPA 1974). [W]hen the prior art teaches away from combining certain known elements, discovery of a successful means of combining them is more likely to be non-obvious. *See KSR v. Teleflex,* 550 U.S. ___, 127 S. Ct. 1727 (2007) *citing United States v. Adams,* 383 U. S. 39, 51-52 (1966). A factfinder should be aware, of course, of the distortion caused by hindsight bias and must be cautious of arguments reliant upon *ex post* reasoning. See *KSR v. Teleflex,* 550 U.S. ___, 127 S. Ct. 1727 (2007) citing Graham v. John Deere Co. of Kansas City, 383 U. S. 1, 36 (warning against a "temptation to read into the prior art the teachings of the invention in issue" and instructing courts to "'guard against slipping into the use of hindsight'" (*quoting Monroe Auto Equipment Co. v. Heckethorn Mfg. & Supply Co.,* 332 F. 2d 406, 412 (CA6 1964))).

The claimed subject matter relates to an attributed debugging system that enables a developer to associate a specific view of an object for examining in a debugger. The specific view includes only object information necessary for the developer to debug the object. The specific view is generated by a display proxy associated with the object. A debugger substitutes the display proxy for the object when examining the object. Independent claim 1 recites *a*

computer-implemented attributed debugging system comprising: a debugger that facilitates debugging of a computer software application, the debugger obtains values of one or more properties of an object of the computer software application; **an expression evaluator, associated with the debugger, that examines a display proxy in place of the object, the display proxy is implemented as a private nested class of the object, the display proxy is configured to expose a subset of the one or more properties of the object, the subset excludes implementation-specific properties of the object;** *and a variable display component that presents, to a developer, values associated with the subset of the one or more properties of the object exposed by the display proxy.* Dandoy fails to teach or suggest such features.

Dandoy relates to a user interface (UI) debugger. Dandoy discloses a debug agent that is combined with a software application. The debug agent collects execution data relating to graphical user interfaces during runtime and proves collected data to a UI debugger. For instance, the debug agent correlates data objects (e.g., instantiated objects associated with a type class) to UI elements in the interface. The debug agent can further monitor the user interface for events. The debug agent provides execution data to a UI debugger or other debugger upon a user request. Moreover, the debug agent can alter properties of UI object during debugging. (*See* paragraphs 18, 21, 22, and 25).

However, Dandoy fails to teach or suggest a display proxy that is examined by an expression evaluator in place of an object. Rather, Dandoy discloses a debug agent associated with an application as a whole and forwards execution data (e.g., properties, events, run-time states, etc.) of user interface objects to a UI debugger. User interfaces and associated software objects are difficult to debug since a majority of UI objects are not in scope during execution of backend application logic. Dandoy discloses a debug agent that extracts properties of UI objects and forwards such information to a UI debugger in order to provide a state of an interface to a user. Thus, Dandoy discloses a UI debugger that obtains properties of UI objects from instantiated UI objects themselves. While a debug agent is employed to extract and forward, the information, the debug agent examines UI object directly and completely.

In the claimed subject matter, an expression evaluator examines a display proxy in place of an object. The display proxy is a private nested class of the object (e.g., a class defined within a class) and exposes a subset of properties of the object. The subset excludes implementation specific properties. Dandoy is silent regarding a display proxy. Rather, the debug agent

examines objects directly. Dandoy nowhere discloses a display proxy that is examined by the debug agent in place of an object.

Independent claim 27 recites, in part, *the expression evaluator examines the display proxy in place of the object to determine debug information that includes values for the relevant properties of the at least one object.* As discussed supra, Dandoy discloses a debug agent that extract execution data of UI objects from the object themselves. Dandoy fails to teach or suggest examining display proxies instead of the objects. In addition, claim 27 recites that the display proxy provides relevant properties regarding a state of the object and conceals properties related to implementation of the object. Dandoy is silent regarding this feature. Dandoy discloses extraction of execution data from objects to enable a UI debugger to recreate a user interface. Accordingly, all properties including implementation related properties must be extracted and forwarded. Moreover still, claim 27 recites an attribute cache directory that retains instances of one or more display proxies, the expression evaluator queries the attribute cache directory for instances of display proxies associated with the at least one object, the expression evaluator creates an instance if not retained in the attribute cache directory. Dandoy discloses extracting execution data associated with UI objects from values stored in memory. Thus, Dandoy extracts execution data from *object* instantiations and not display proxies. Nowhere does Dandoy disclose creating or retaining display proxies.

In view of at least the foregoing, it is readily apparent that Dandoy fails to teach or suggest the claimed invention as recited in independent claims 1 and 27 (and associated claims that depend therefrom). Accordingly, this rejection should be withdrawn and the claims allowed.

II.   **Rejection of Claims 19 and 20 Under 35 U.S.C. §103(a)**

Claims 19 and 20 stands rejected under 35 U.S.C. §103(a) over Dandoy in view of Bates et al. (U.S. Patent 6,961,924). It is respectfully requested that this rejection be withdrawn for at least the following reason. Claims 19 and 20 depend from independent claim 1 and Bates et al. does not rectify the deficiencies presented by Dandoy with respect to independent claim 1, as discussed above. Accordingly, withdrawal of this rejection is respectfully requested.

III.   Rejection of Claims 21-23 and 26 Under 35 U.S.C. §103(a)

Claims 21-23 and 26 stands rejected under 35 U.S.C. §103(a) over Bates et al. in view of Dandoy. It is respectfully requested that this rejection be withdrawn for at least the following reasons. Bates et al. and Dandoy, either alone or in combination, do not teach or suggest all limitations recited in the subject claims.

Independent claim 21 recites, in part, *examining the display proxy in place of the object to determine debug information related to the object.* Similarly, independent claim 26 recites, in part, *means for examining the display proxy in place of the object, means for examining the display proxy comprise means for obtaining values of the provided relevant properties.* The cited art fails to teach or suggest such features.

Rather, Bates et al. relates to debugging software code wherein the debugger includes additional descriptive material associated with variables beyond the values stored by the variables. In one embodiment, comments associated with a variable are displayed in a debugger in connection with the variable. The comments can be external (e.g., stored in a database) or internal (e.g., comments within source code). (See col. 3, ll. 39-51). In addition, information can be provided that describes the use of a variable. For example, attributes can be associated with variables that indicate the variable is a global, static, an index value, a parameter, a return value or a call value. (See col. 3, ln. 64 – col. 4, ln. 4). During the debugging process, the debugger checks to see if any attributes are associated with a variable and displays the attribute accordingly. (See col. 11, ll. 32-50). Thus, Bates et al. relates to providing information, such as comments or use information, in addition to the value of variables.

In the claimed subject matter, a display proxy, implemented as a private nested class of an object, is employed to generate debug information related to the object (e.g. a class variable or the like). The display proxy is examined by the debugger *in place of the object.* As recited in the subject claims, a display proxy provides relevant features of an object while concealing implementation specifics. For example, consider a HashTable object wherein keys are utilized in connection with values to store values in the HashTable. While debugging, a developer may only be interested in the keys and associated values of the HashTable. A debugger examining the actual object will reveal details that include complex variables specifying how the HashTable implements its functionality. A display proxy, however, will reveal the keys and values in a meaningful manner when examined in place of the object and conceal the complex variables

relating to implementation. Bates et al. discloses examining variables themselves and not a proxy that only provides relevant features and not implementation specifics. For example, Fig. 7 of Bates et al. shows a variable "name" that includes both a value of a pointer as well as the value of the field pointed thereto. Thus, Bates et al. reveals the entirety of a variable. Further, Bates et al. is silent regarding examining a proxy that is a private nested class of an object to obtain variable value information.

Dandoy is relied upon to cure the aforementioned deficiencies of Bates *et al.* with respect to the claims. As discussed *supra*, Dandoy discloses a debug agent that is association with an entirety of an application and acts as a point of entry for execution. The debug agent monitors and/or collects data related to all user interface objects in the application. In the claimed subject matter, the display proxy is a private nested class of an object and, accordingly, has a one-to-one association. In addition, the display proxy is examined for information (e.g., object properties, etc.). In Dandoy, the objects themselves are examined to determine execution information. Thus, Dandoy fails to make up for the deficiencies of Bates et al. with respect to the subject claims.

In view of at least the foregoing, it is readily apparent that Bates et al. and Dandoy, alone or in combination, fail to teach or suggest the claimed invention as recited in independent claims 21 and 26 (and associated claims that depend therefrom). Accordingly, this rejection should be withdrawn and the claims allowed.

CONCLUSION

The present application is believed to be in condition for allowance in view of the above comments and amendments. A prompt action to such end is earnestly solicited.

In the event any fees are due in connection with this document, the Commissioner is authorized to charge those fees to Deposit Account No. 50-1063 [MSFTP579US].

Should the Examiner believe a telephone interview would be helpful to expedite favorable prosecution, the Examiner is invited to contact applicants' undersigned representative at the telephone number below.

Respectfully submitted,

AMIN, TUROCY & CALVIN, LLP

 /Evan T. Perry/

Evan T. Perry

Reg. No. 62,190

AMIN, TUROCY & CALVIN, LLP
127 Public Square
57th Floor, Key Tower
Cleveland, Ohio 44114
Telephone (216) 696-8730
Facsimile (216) 696-8731